



360Store[®]

Back Office Administrator Guide

Confidential

©2005 360Commerce. All rights reserved. 360Commerce and third-party specifications are subject to change without notice. Although every precaution has been taken in the preparation of this paper, 360Commerce assumes no responsibility for errors or omissions, and no warranty or fitness is implied. 360Commerce shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the use of the information contained in this paper. Trademarks: 360Commerce (both word and logo), 360Store, 360Enterprise, Unleashed, Warm-Start Optimization. All terms mentioned in this paper that may be trademarks or service marks have been appropriately capitalized or otherwise designated.

This document and the information it contains are the property of 360Commerce, Inc. This document contains confidential and proprietary information that is not to be disclosed to other parties. The information in this document may not be used by other parties except in accordance with a written agreement signed by an officer of 360Commerce, Inc.

TABLE OF CONTENTS

Preface ix

Chapter 1: Preparing to Deploy

- Hardware Requirements 1-1
- Software Requirements 1-2
- Printer Requirements 1-2
- Supported Products 1-3
- Installing and Setting Up the Database 1-3

Chapter 2: Deploying On JBoss

- Copying Application and Utility Files 2-1
- Configuring Files 2-3
 - Setting Recursive Searching 2-3
 - Changing JVM Heap Size 2-3
 - Configuring Mail Services 2-4
 - Configuring Data Sources for JBoss 2-5
 - Configuring Quartz 2-6
 - Setting the Store Number 2-6
- Configuring Queue Destinations 2-7
 - Configuring a Store Queue 2-8
 - Configuring the parameters Topic 2-8

Chapter 3: Configuring WebSphere MQ and WebSphere and Deploying Back Office

- Configuring WebSphere MQ 3-9
 - Configuring Queue Destinations 3-10
 - Creating a Queue Manager and Queues for SUSE Linux 3-11
 - Creating the Queue Manager 3-11
 - Creating Queues Manually 3-11
 - Creating a Queue Manager and Queues for Windows 3-12
 - Creating the Queue Manager 3-12
 - Connecting to the Queue Manager Using MQ Explorer 3-14
 - Creating an Error Queue using MQ Explorer 3-15
 - Creating a Local Queue using MQ Explorer 3-16
 - Creating Multiple Queues Using an Input Data File 3-18
- WebSphere Resources Required by Back Office 3-20
 - Other Requirements 3-21
- Configuring WebSphere Application Server (WAS) 3-22
 - Configuration Using Scripts 3-22
 - Configuring the WAS Properties 3-22
 - Building the Security Jar 3-23
 - Configuring WebSphere and Back Office Security 3-23
 - Configuring Additional Server Configuration for Back Office 3-24
 - Manual Configuration 3-24
 - Application Servers 3-24
 - Resources 3-26
 - Security 3-32

- Copying Files 3-35
- Deploying Back Office 3-36
 - Installing Back Office 3-36
 - Restarting the WebSphere Application Server 3-36

Chapter 4: Configuring the AccessVia Print Engine for Labels and Tags

- Configuring the Database for the AccessVia Print Engine 4-2
 - Configuring DB2 4-2
 - Configuring MySQL 4-3
- Setting Up the Directory Structure 4-4
 - Setting up the Linux Directory Structure 4-4
 - Setting up the Windows Directory Structure 4-4
- Creating the AccessVia Print Engine .ini File 4-5
 - .ini File Settings 4-5
 - .ini File Prototype 4-6
- Setting the Database Pointer 4-9
 - Setting the Database Pointer for WebSphere 4-9
 - Setting the Database Pointer for JBoss 4-9
- Setting Variables for the AccessVia Print Engine 4-9
 - Setting Variables for WebSphere 4-9
 - Setting Variables for JBoss 4-10
- Testing Standalone Printing 4-10
 - Testing AccessVia in WebSphere 4-10
 - Testing AccessVia in JBoss 4-10
- Troubleshooting 4-11
 - MalformedURLException 4-11
 - ClassNotFoundException 4-11

Chapter 5: Configuring Back Office

- Loading Labels and Tags 5-1
- Configuring Security 5-1
- Starting Up Back Office 5-2
- Managing Parameters 5-2
- Scheduling Post-processors 5-3

LIST OF TABLES

Table P-1	Text Conventions	x
Table 1-1	Minimum Hardware Requirements	1-1
Table 1-2	Supported Products	1-3
Table 2-1	Back Office Application and Utility Files	2-1
Table 2-2	Files Requiring Configuration	2-3
Table 2-3	Required JMS Queues for the Store JMS Server	2-7
Table 3-1	Typical JMS Queues for the Store JMS Server	3-10
Table 3-2	Server Custom Properties	3-25
Table 3-3	Example: Message Listener Service Configuration	3-26
Table 3-4	Listener Ports	3-26
Table 3-5	Example: JDBC XA Provider	3-27
Table 3-6	Example: JDBC XA Data Source	3-27
Table 3-7	Example: JDBC XA Crystal Clear Data Source	3-28
Table 3-8	Example: DB2 JDBC Thin Driver	3-29
Table 3-9	Example: JDBC (non-XA) DataSource Configuration	3-29
Table 3-10	Example: MQ Queue Destination Options	3-31
Table 3-11	Typical Queues	3-32
Table 3-12	Example: MQ Queue Destination Options	3-32
Table 3-13	Example: User Registry Security Configuration	3-33
Table 3-14	Example: Global Security Configuration	3-34
Table 3-15	Example: J2C Authentication Data for XA Resource	3-34
Table 3-16	Example: J2C Authentication Data for non-XA resource	3-35
Table 3-17	Example: Files to Copy	3-35

LIST OF FIGURES

Figure 3-1	Create Queue Manager (Step 1) Screen	3-12
Figure 3-2	Create Queue Manager (Step 2) Screen	3-13
Figure 3-3	Create Queue Manager (Step 3) Screen	3-13
Figure 3-4	Create Queue Manager (Step 4) Screen	3-14
Figure 3-5	MQ Explorer: Queues Folder	3-15
Figure 3-6	Create Local Queue Window: Error Queue	3-16
Figure 3-7	MQ Explorer: Queues Folder	3-16
Figure 3-8	Create Local Queue Window	3-17
Figure 3-9	Create Local Queue: Enter Backout Requeue Name	3-18

Audience

This document is intended for system and application administrators who are responsible for installing, deploying, configuring, and administering Back Office in a production environment.

About this Book

This book describes how to install, deploy, configure, and administer the Back Office application.

Feedback

Please e-mail feedback about this document to 360University@360Commerce.com.

Trademarks

The following trademarks and copyrights may be found in 360Commerce® documentation:

- 360Commerce, 360Store and 360Enterprise are registered trademarks of 360Commerce Inc.
- Unleashed is a trademark of 360Commerce Inc.
- IBM is a registered trademark of International Business Machines Corporation in the United States, other countries, or both.
- Windows is a registered trademark of Microsoft Corporation.
- Java is a trademark of Sun Microsystems Inc.
- Linux is a registered trademark of Linus Torvalds.
- The Jetty Package is copyright (c) Mort Bay Consulting Pty. Ltd. (Australia) and others.

All other trademarks mentioned herein are the properties of their respective owners.

Text Conventions

The following table shows the text conventions used in this document.

Table P-1 Text Conventions

Sample	Description
<i>Italic text</i>	This is used for emphasis. It calls attention to crucial information and important terms defined in the text, for example, <i>quantity on hand</i> is....
Bold text	This is used for text (in an application window or on a keyboard) that is acted upon by the user, for example, Click Next .
Courier Text	This is used for code, including file and directory names, paths, syntax, and commands, for example, <code>/opt/jboss-3.2.1/server/default/lib</code> .
< <i>Italics and angle brackets</i> >	This is used for text that needs to be supplied by the user, for example, Set the parameter <code>AccessViaIniFilePath = /opt/accessvia/program/<AccessVia ini file name></code> . If it is within a code sample, the text is in Courier font.

PREPARING TO DEPLOY

This document provides a general summary of the deployment process and describes the prerequisites you must satisfy prior to deploying Back Office. Actual deployment steps vary for different application servers and databases; therefore, this guide is intended to be used with the product documentation provided with your database, application server, and other software.

The deployment of Back Office requires the following steps:

- Acquiring and configuring the necessary hardware. See “Hardware Requirements.”
- Acquiring and installing the prerequisite software. See “Software Requirements.”
- Setting up the database. See “Installing and Setting Up the Database.”
- Configuring and starting the messaging service and application server. See “Deploying On JBoss” and “Configuring WebSphere MQ and WebSphere and Deploying Back Office.”
- Loading the necessary parameters. See “Managing Parameters.”

Hardware Requirements

Specific hardware requirements for running Back Office depend on several variables, including the choice of operating system, middleware, the number of stock-keeping units (SKUs), the transaction volume, and the deployment architecture. 360Commerce consultants will work with you to determine the ideal configuration for your deployment of Back Office.

Following are the minimum hardware specifications for deploying Back Office:

Table 1-1 Minimum Hardware Requirements

Hardware	Minimum Requirement
System CPU	1 GHz Pentium-class processor
RAM	1 GB
Hard-disk space	10 GB

In addition, in order to use Unleashed Inventory or Labels and Tags, your system requires an appropriate handheld device with an embedded operating system.

Software Requirements

Back Office is packaged as an Enterprise Archive (.ear) file and must be deployed to a J2EE 1.3-compliant application server. 360Commerce software is designed to open standards and is compatible with many operating systems, application servers, databases, and other third-party software.

The Back Office application requires that the following software be installed and configured:

- A Java environment (J2EE) v. 1.3.x
- Java Development Kit (JDK) v. 1.4.x
- A J2EE 1.3-compliant application server
- Messaging software
- A database server

If you plan to use SQL scripts to create your database, you must also install appropriate software to execute the scripts; Apache Ant is recommended. For information about Apache Ant, see <http://ant.apache.org>.

Printer Requirements

In order to use the Labels and Tags functionality of Back Office, your system requires the AccessVia Print engine and a compatible printer. For printers supported by AccessVia, see the AccessVia documentation. For the necessary configuration for the AccessVia Print engine, see Chapter 4, “Configuring the AccessVia Print Engine for Labels and Tags.”

Supported Products

360Commerce applications are open and can run on many configurations comprised of various hardware and software components. For this release, the following products are supported.

Table 1-2 Supported Products

Component	Supported Products
Operating System	IRES V2 (SUSE Linus Enterprise Server 9) Windows 2003 Server
JDK	1.4.2
Application Server	WebSphere Remote Server (WRS) 5.1.2 (5.1.1.3) JBoss 4.0.2
Messaging Software	WebSphere MQ Series 5.3 Fix Pack 11 JBoss JMS 4.0.2
Database	DB2 v8.2 MySQL 4.1.14
Hardware	Dell 2650 Dell 2850

Installing and Setting Up the Database

The installation and configuration of a database server is a prerequisite for deploying Back Office. 360Commerce recommends that you have a qualified DBA install and configure your database prior to installing Back Office and its database schemas.

The easiest way to install the Back Office database is by using Apache Ant. In the `backofficeDBInstall.jar` file, 360Commerce provides SQL scripts to create the database and a `build.xml` file that identifies the Ant target.

To install the database using 360Commerce SQL scripts:

1. Unjar `backofficeDBInstall.jar` into a directory.
2. Run the following script from the directory:

- `ant load_sql`

This creates the database tables and installs sample data.

DEPLOYING ON JBOSS

This chapter provides a list of steps that a typical JBoss administrator needs to perform in order to deploy Back Office. This consists of two basic steps:

- Copy the necessary files to the JBoss installation directory. See “Copying Application and Utility Files.”
- Configure the necessary files. See “Configuring Files.”

Copying Application and Utility Files

Locate the appropriate build of Back Office in the source code control system and copy files from there to your JBoss installation directory. The provided files include not only the Back Office application itself but also a series of utilities bundled with the application and used to facilitate its operation. The source paths listed in the following table are examples; actual paths could vary in your deployment

Table 2-1 Back Office Application and Utility Files

File	From path (relative to 360Store Back Office build in StarTeam)	To path (relative to JBoss installation directory)	Description
backoffice.ear	\applications\backoffice\assemble\assemble.working.dir	\server\default\deploy	Back Office enterprise archive file
log4jxxx.jar (log4j-1.2.8.jar)	\thirdparty\apache	\server\default\lib	Jakarta logging utility, where xxx represents the version; because JBoss comes with Log4J, this is only necessary if your installation requires a different version of Log4J
db2jcc.jar db2jcc_license_cisuz.jar db2jcc_license_cusuz.jar	\thirdparty\db2	\server\default\lib	Database driver for DB2; all three of these files must be the same version

Table 2-1 Back Office Application and Utility Files

File	From path (relative to 360Store Back Office build in StarTeam)	To path (relative to JBoss installation directory)	Description
mysql-connector-java-3.1.10-bin.jar	\thirdparty\mysql		Database driver for MySQL
quartz.properties	\applications\backoffice\appservers\jboss\3.2.1\server\default\conf	server\default\conf	Quartz configuration file
log4j.xml			Editable Log4J configuration file
login-config.xml			Security configuration file; needed only if you are using 360Commerce default security
jbossmq-destinations-service.xml	\applications\backoffice\appservers\jboss\3.2.1\server\default\deploy\jms	\server\default\deploy\jms	Default JMS queue destinations for Back Office; see “Setting the Store Number” on page 2-6 for more information
backoffice-default.war	\applications\backoffice\appservers\jboss\3.2.1\server\default\deploy	\server\default\deploy\	Configures JBoss to send loss-of-context errors to Back Office; necessary to fix a bug in Jetty, an HTTP server with servlet capabilities and associated utilities. Jetty is shipped with JBoss 3.2.1
mail-service.xml			Mail service configuration file
<dbname>-ds.xml			Data source file
hsqldb-ds.xml			Data source file

Configuring Files

The following table lists the files that require configuration for deploying Back Office in JBoss.

Table 2-2 Files Requiring Configuration

File	Required Configuration
jboss-services.xml	Set/unset recursive searching
run.sh	Increase the heap size
mailservice.xml	Set mail service properties
<dbname>-ds.xml	Set data source for JBoss
hsqldb-ds.xml	Set data source for JBoss
quartz.properties	Set data source and persistence
ejb-jar.xml	Set the store number
jbossmq-destinations-service.xml	Add/remove queues

Setting Recursive Searching

By default, recursive searching is set to `true` in JBoss; consequently, it is necessary to configure this property only if you want to turn off recursive searching or restore it after it has been turned off.

To turn on recursive searching in JBoss:

1. Open the file `<JBoss installation>\server\default\conf\jboss-service.xml` in a text editor.
2. Within the `<mbean>` tag, locate the `RecursiveSearch` attribute and set its value to `true`.

```
<attribute name="RecursiveSearch">True</attribute>
```

Changing JVM Heap Size

If you require a larger than default JVM heap, use one of the following procedures:

To increase the JVM heap size for Windows:

1. Open `<JBoss installation>\run.bat` in edit mode.
2. In `run.bat`, locate the following lines:

```
Sun JVM memory allocation pool parameters  
set JAVA_OPTS=%JAVA_OPTS%-Xms128m -Xmx512m
```

where 128 MB is the minimum size and 512 MB is the maximum size.
3. Change the final parameter to the heap size you want to enable.

To increase the JVM heap size for Linux:

1. Open `<JBoss installation>\run.conf` in edit mode.
2. In `run.conf`, locate the following lines:

Specify options to pass to the Java VM

```
set JAVA_OPTS="-server -Xms128m -Xmx512m
```

where 128 MB is the minimum size and 512 MB is the maximum size.

3. Change the final parameter to the heap size you want to enable.

Configuring Mail Services

Mail services enable you to receive notification of events, for example, the performance of pre-arranged jobs, that have been executed by the application server.

To configure mail services for Back Office:

1. Open the `<JBoss installation>\server\default\deploy\ mail-service.xml` file. Following is an example of that file.

```
<!-- ===== -->
<!-- Mail Connection Factory -->
<!-- ===== -->

<mbean code="org.jboss.mail.MailService"
      name="jboss:service=Mail">
  <attribute name="JNDIName">java:/Mail</attribute>
  <!--
    <attribute name="User">nobody</attribute>
    <attribute name="Password">password</attribute>
  -->
  <attribute name="Configuration">
    <!-- Test -->
    <configuration>
      <!-- Change to your mail server protocol -->
      <property name="mail.store.protocol" value="pop3"/>
      <property name="mail.transport.protocol" value="smtp"/>

      <!-- Change to the user who will receive mail
      <property name="mail.user" value="nobody"/>-->
```

```

        <!-- Change to the mail server -->
        <property name="mail.pop3.host" value="360cmail"/>

        <!-- Change to the SMTP gateway server -->
        <property name="mail.smtp.host" value="360cmail"/>

        <!-- Change to the address mail will be from -->
        <property name="mail.from" value="no_reply@360commerce.com"/>

        <!-- Enable debugging output from the javamail classes -->
        <property name="mail.debug" value="false"/>
    </configuration>
</attribute>
</mbean>
</server>

```

2. Edit the relevant attributes as necessary, including mail protocol, mail server, SMTP server, gateway, and return address. Your system administrator should be able to provide that information.

Configuring Data Sources for JBoss

Connect JBoss to the appropriate data sources for your deployment by editing files in the *<JBoss installation>\server\default\deploy* directory. In a system that uses the two standard data sources, edit the *<databasename>-ds.xml* and *hsqldb-ds.xml* files.

To configure the data sources:

1. In the *<databasename>-ds.xml* file, configure the *<datasources>* section to include the following:
 - A data source with the JNDI name *jdbc/DataSource*
 - An appropriate *<connection-url>* tag

An example follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<datasources>
    <local-tx-datasource>
        <jndi-name>jdbc/DataSource</jndi-name>
        <connection-url><--database-specific connection URL--></connection-url>
    
```

```

        <driver-class><--fully qualified driver class name--></driver-class>

        <user-name><--database user name--></user-name>

        <password><--corresponding password--></password>

    </local-tx-datasource>

</datasources>

```

2. Copy the changes you made to the `<datasenname>-ds.xml` to the `hsqldb-ds.xml` file.

Configuring Quartz

Quartz is an open source job scheduling system that can be integrated or used with a J2EE application. For information about Quartz, see the Quartz documentation at <http://www.quartzscheduler.org/quartz/>. This release of Back Office uses Quartz version 1.3.

To configure Quartz:

Edit the `quartz.properties` file data sources to match the data sources you are using with Back Office and to make Quartz store its data in a database rather than in memory.

In the following code sample, two lines define the two standard data sources:

```

org.quartz.dataSource.DEFAULT.jndiURL = java:/jdbc/DataSource
org.quartz.dataSource.ALT.jndiURL = java:/jdbc/Other

```

Setting the Store Number

Each store has a unique number that connects it to Central Office. This number must be set in the `ejb-jar.xml` file.

To set the store number:

1. Open the `backoffice.ear` file. It contains the `storeserver-app-ejb.jar` file.
2. Open the `storeserver-app-ejb.jar` file. It contains the `ejb-jar.xml` file.
3. Locate the following section of the `ejb-jar.xml` file:

```

<enterprise-beans>

    <message-driven id="MessageDriven_ParameterImportAndForwardServiceBeanMDB">

        ...

        <env-entry>

            <env-entry-name>storeID</env-entry-name>

            <env-entry-type>java.lang.String</env-entry-type>

            <env-entry-value>04242</env-entry-value>

        </env-entry>

        ...

```

</message-driven>

</enterprise-beans>

4. Change the value of `<env-entry-value>` to the correct store number.

Configuring Queue Destinations

Your Back Office deployment requires a set of configured JMS queues and topics on the store JMS server. A *queue* is a point-to-point message conduit. A *topic*, on the other hand, handles messages to be sent to multiple subscribers. The actual list of queues and topics required for your deployment can vary depending on the options you purchase and how you configure Back Office. Determine the required configuration for your deployment with your development team and configure the queues as appropriate for your application server. A variety of standard queues are required, along with one queue per store. Refer to the following table for a list of the required queues.

Table 2-3 Required JMS Queues for the Store JMS Server

Name	JNDI Name
Acknowledgement	jms/Acknowledgement
EJournal	jms/EJournal
EJournalImport	jms/EJournalImport
ItemImport	jms/ItemImport
POSLog	jms/POSLog
PricingImport	jms/PricingImport
AuditLogExport	jms/AuditLogExport
ParameterImport	jms/ParameterImport
SignatureCapture	jms/SignatureCapture
Store XXXXX	jms/store_XXXXX; see “Configuring a Store Queue” on page 2-8.
TaskExecution	jms/TaskExecution
TaxImport	jms/TaxImport
JournalingMessage	jms/JournalingMessage

Edit the `<JBoss installation>\server\default\deploy\jms\jbossmq-destinations-service.xml` file to add and remove queues for your deployment. To add a queue, copy and edit `<mbean>` tags within the XML file.

Configuring a Store Queue

Each store must have a queue defined in the format `store_<XXXXXX>`, where `<XXXXXX>` represents the unique store number. This number is identical to the number you set in the `ejb-jar.xml` file. See “Setting the Store Number” on page 2-6.

The following code sample illustrates the configuration for a store queue:

```
<mbean code="org.jboss.mq.server.jmx.Queue"
      name="jboss.mq.destination:service=Queue,name=store_00001">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

Configuring the parameters Topic

Configure the `parameters` topic, which is also handled by an `<mbean>` tag. The `parameters` topic handles the distribution of parameters. For an explanation of parameters, see “Managing Parameters” on page 5-2. The following code sample illustrates the default values for a `parameters` topic:

```
<mbean code="org.jboss.mq.server.jmx.Topic"
      name="jboss.mq.destination:service=Topic,name=parameters">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
  <depends optional-attribute-
name="SecurityManager">jboss.mq:service=SecurityManager</depends>
  <attribute name="SecurityConf">
    <security>
      <role name="guest" read="true" write="true" create="true"/>
      <role name="publisher" read="true" write="true" create="true"/>
      <role name="durpublisher" read="true" write="true" create="true"/>
    </security>
  </attribute>
</mbean>
```

CONFIGURING WEBSPHERE MQ AND WEBSPHERE AND DEPLOYING BACK OFFICE

This chapter provides the procedures necessary to configure WebSphereMQ, to configure WebSphere, and to deploy Back Office into WebSphere. This chapter does not explain how to install WebSphere. See your WebSphere documentation or go to <http://www.redbooks.ibm.com> for the most current installation instructions.

This chapter includes the following topics:

- “Configuring WebSphere MQ”
- “WebSphere Resources Required by Back Office”
- “Configuring WebSphere Application Server (WAS)”
- “Deploying Back Office”

Configuring WebSphere MQ

The following sections describe how to create and use a queue manager for Back Office and how to define the queues required by Back Office. There are different procedures for SUSE Linux and Windows. In addition, you can create queues manually, one at a time, or you can create multiple queues using a data input file. See the following topics for instructions:

- “Configuring Queue Destinations”
- “Creating a Queue Manager and Queues for SUSE Linux”
- “Creating a Queue Manager and Queues for Windows”
- “Creating Multiple Queues Using an Input Data File”

Configuring Queue Destinations

Your Back Office deployment requires a set of configured JMS queues and topics on the store JMS server. A *queue* is a point-to-point message conduit. A *topic*, on the other hand, handles messages to be sent to multiple subscribers. The actual list of queues and topics required for your deployment may vary depending on the options you purchase and your configuration of Back Office. A variety of standard queues are required, along with one queue per store. Refer to the following table for a list of typical queues.

Table 3-1 Typical JMS Queues for the Store JMS Server

Name	JNDI Name
Acknowledgement	jms/Acknowledgement
EJournal	jms/EJournal
EJournalImport	jms/EJournalImport
ItemImport	jms/ItemImport
POSLog	jms/POSLog
PricingImport	jms/PricingImport
AuditLogExport	jms/AuditLogExport
ParameterImport	jms/ParameterImport
SignatureCapture	jms/SignatureCapture
Store XXXXX*	jms/store_XXXXX*
TaskExecution	jms/TaskExecution
TaxImport	jms/TaxImport
* Each store must have a queue defined; XXXXX represents the store number. Use the format shown, as Back Office expects this format.	

An example JMS topology is contained in the `/root/install/<build number>/backofficeInstall/appservers/was/createq.dat` file. This file is designed to work with the test data provided by 360Commerce and installed by `ws_ant`. Your WebSphere MQ administrator must modify this topology as necessary for your environment.

After modifying the JMS topology, create a queue manager and configure it. The following section provides procedures for creating a simple queue manager and its topology.

Creating a Queue Manager and Queues for SUSE Linux

The following sections describe how to create a queue manager for Back Office and how to define the error and normal queues required by Back Office. For a list of required queues, see Table 3-1, “Typical JMS Queues for the Store JMS Server” on page 3-10.

Creating the Queue Manager

Use the following procedure to create a queue manager. Before doing this, confirm that no other processes use Port 1414.

To create the queue manager:

1. In the mq installation bin directory (`$mq_base\bin`), execute the following command to create the queue manager:

```
/opt/mqm/bin > crtmqm -q bo.queue.manager
```

2. Start the queue manager and queue listener using the following commands:

```
/opt/mqm/bin > strmqm bo.queue.manager
```

```
/opt/mqm/bin > runmqslsr -m bo.queue.manager -p 1414 -t tcp
```

Creating Queues Manually

If you need to create a queue manually, use the following procedure. For more detailed instructions, see the WebSphere MQ documentation. To create multiple queues using an input data file, see “Creating Multiple Queues Using an Input Data File” on page 3-18.

To create a queue manually:

1. If not already opened, enter the command line to open an mq shell:

```
mqm@sunrise:/opt/mqm/bin> runmqsc -m bo.queue.manager
```

2. Enter a command in the mq shell to create the queue:

- If the local queue does not need an error queue, execute the following command:

```
DEFINE QLOCAL(EJOURNALIMPORT) USAGE (NORMAL) DEFPSIST(YES)
```

- If the local queue does need an error queue (the backout queue), use the following command:

```
DEFINE QLOCAL(EJ.ERROR) USAGE (NORMAL) DEFPSIST(YES) +
```

```
DEFINE QLOCAL (EJOURNALIMPORT) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(EJ.ERROR)
```

```
BOTHRESH(1)
```

In this command, the `BOQNAME` parameter is the Backout Queue; the `BOTHRESH` is the Backout Threshold (the number of times the message is tried before moving to the Backout Queue).

3. Wait for the response WebSphere MQ queue created.

Creating a Queue Manager and Queues for Windows

The following sections describe how to create a queue manager for Back Office, how to connect to the queue manager, and how to define the error and normal queues required by Back Office. For a list of required queues, see Table 3-1, “Typical JMS Queues for the Store JMS Server” on page 3-10.

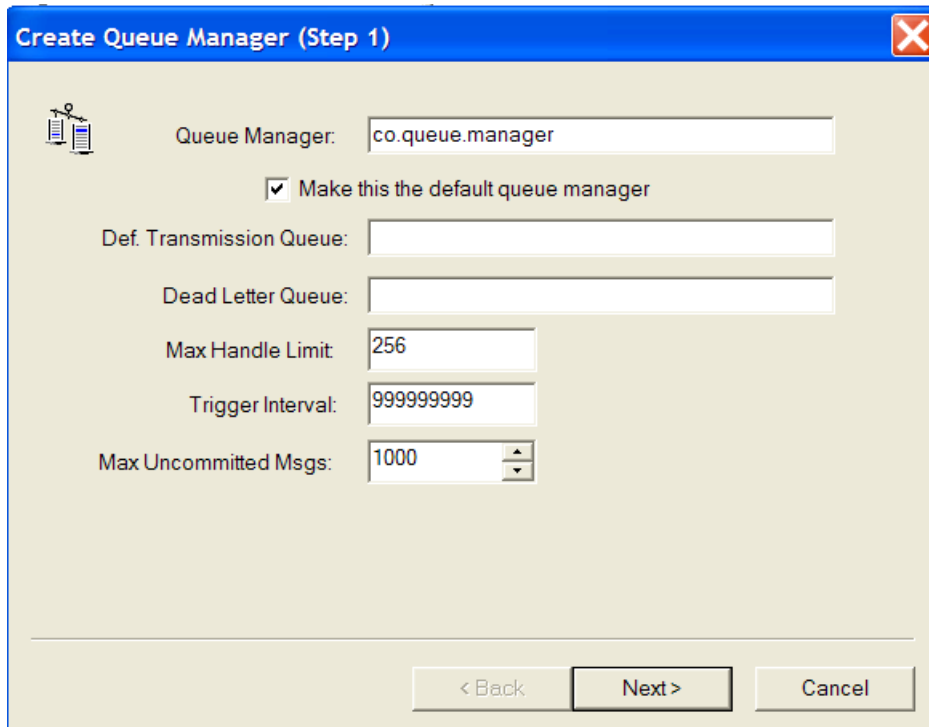
Creating the Queue Manager

Use the following procedure to create a queue manager. Before doing this, confirm that no other processes use Port 1414.

To create the queue manager:

1. Open the MQ Explorer.
2. Expand the WebSphere MQ tree in the left pane of the screen.
3. Right-click the Queue Managers folder in the left pane.
4. Choose **New > Queue Manager**.
5. In the Create Queue Manager (Step 1) screen, enter the name for the Queue Manager .
6. Check **Make this the default queue manager**.
7. Keep the defaults for the remaining fields.

Figure 3-1 Create Queue Manager (Step 1) Screen



Create Queue Manager (Step 1)

Queue Manager:

☒ Make this the default queue manager

Def. Transmission Queue:

Dead Letter Queue:

Max Handle Limit:

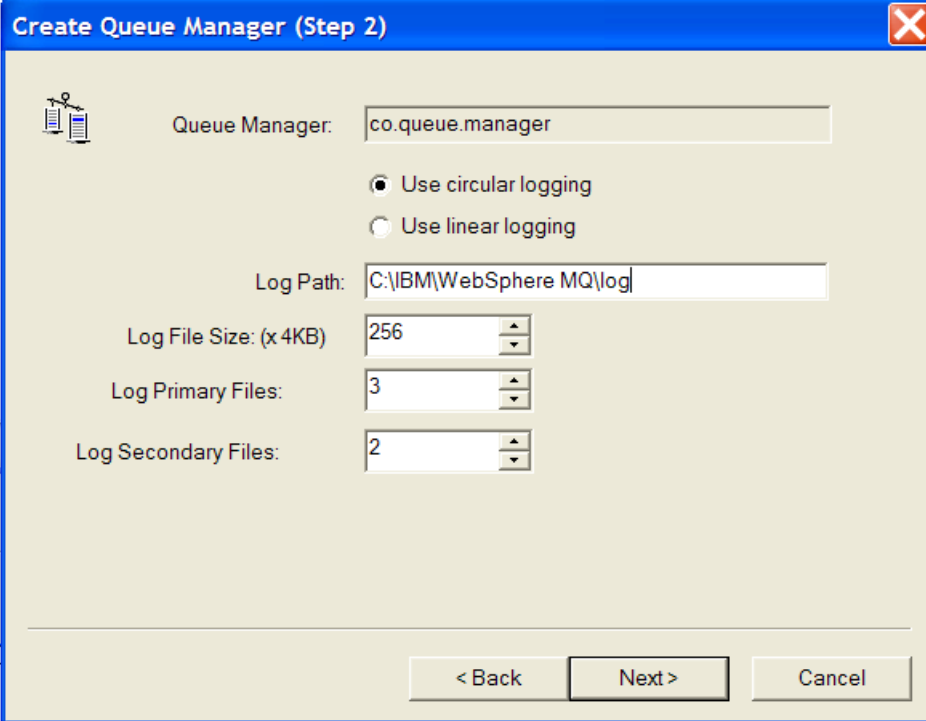
Trigger Interval:

Max Uncommitted Msgs:

< Back Next > Cancel

8. Click **Next**.

Figure 3-2 Create Queue Manager (Step 2) Screen



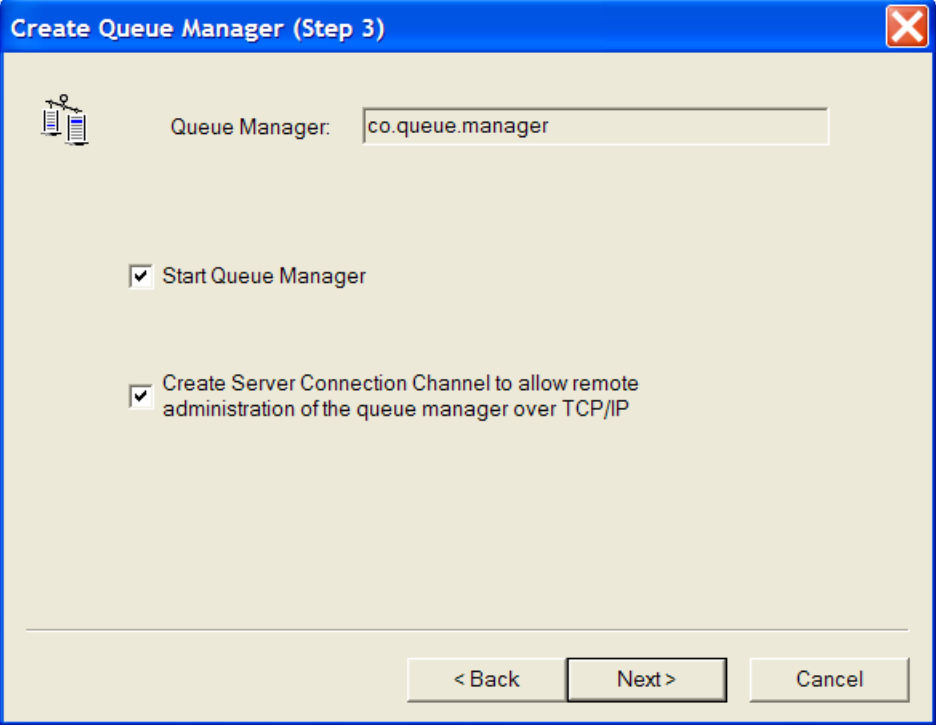
The dialog box titled "Create Queue Manager (Step 2)" contains the following fields and options:

- Queue Manager:** A text field containing "co.queue.manager".
- Logging:** Two radio buttons: "Use circular logging" (selected) and "Use linear logging".
- Log Path:** A text field containing "C:\IBM\WebSphere MQ\log".
- Log File Size: (x 4KB):** A spin box set to "256".
- Log Primary Files:** A spin box set to "3".
- Log Secondary Files:** A spin box set to "2".
- Navigation:** Buttons for "< Back", "Next >", and "Cancel".

9. In the Create Queue Manager (Step 2) screen, click **Next** to accept the defaults.

10. In the Create Queue Manager (Step 3) screen, check both check boxes.

Figure 3-3 Create Queue Manager (Step 3) Screen



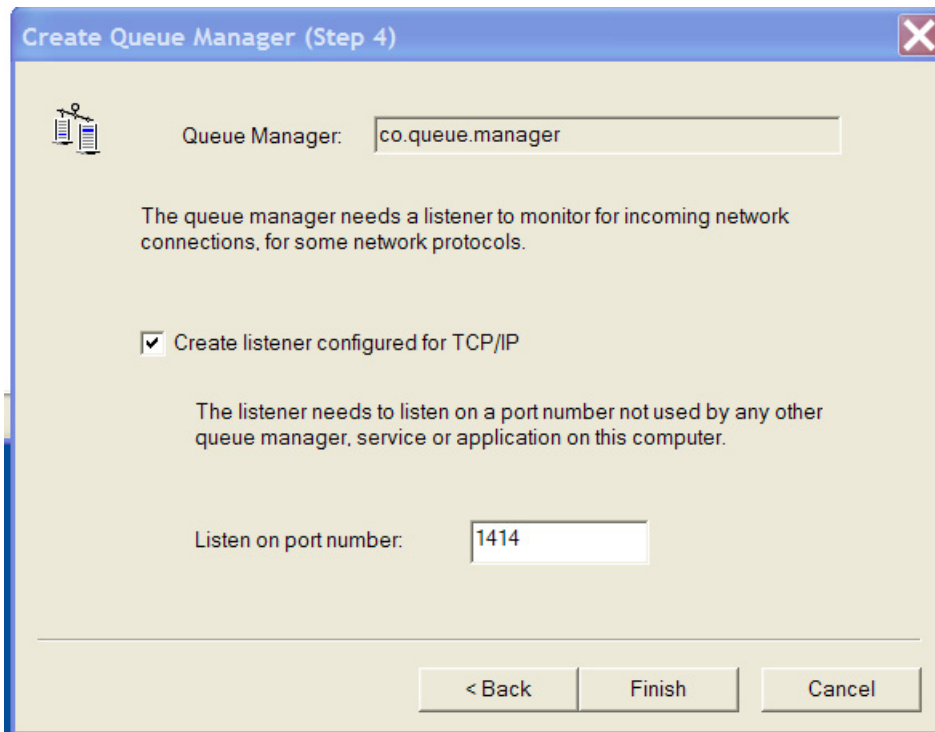
The dialog box titled "Create Queue Manager (Step 3)" contains the following fields and options:

- Queue Manager:** A text field containing "co.queue.manager".
- Start Queue Manager:** A checked checkbox.
- Create Server Connection Channel to allow remote administration of the queue manager over TCP/IP:** A checked checkbox.
- Navigation:** Buttons for "< Back", "Next >", and "Cancel".

11. Click **Next**.

12. In the Create Queue Manager (Step 4) screen, check **Create listener configured for TCP/IP**.

Figure 3-4 Create Queue Manager (Step 4) Screen



Create Queue Manager (Step 4)

Queue Manager:

The queue manager needs a listener to monitor for incoming network connections, for some network protocols.

☒ Create listener configured for TCP/IP

The listener needs to listen on a port number not used by any other queue manager, service or application on this computer.

Listen on port number:

< Back Finish Cancel

13. Enter 1414 in the Listen on port number field.

14. Click **Finish**.

Connecting to the Queue Manager Using MQ Explorer

Use the following procedure to connect to the queue manager using MQ Explorer. If you get a security exception, consult the WebSphere MQ documentation for how to resolve the security settings of the MQ Manager.

To connect to the queue manager:

1. Open the MQ Explorer.
2. Expand the WebSphere MQ tree in the left navigation area of the window.
3. Right-click the Queue Managers folder in the left navigation area.
4. Select **Show, Queue Manager**.
5. Select **Show a remote Queue Manager**.
6. Enter the queue manager name, for example, `bo.queue.manager`.
7. Enter the connection name, for example, `bs1(1414)`.

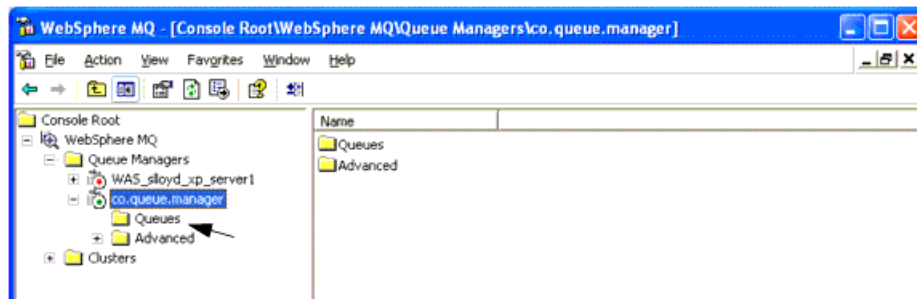
Creating an Error Queue using MQ Explorer

The creation of an error queue is optional. For each queue and error queue pair, create the error queue first, so that it can be selected as the error queue for the normal queue.

To create an error queue:

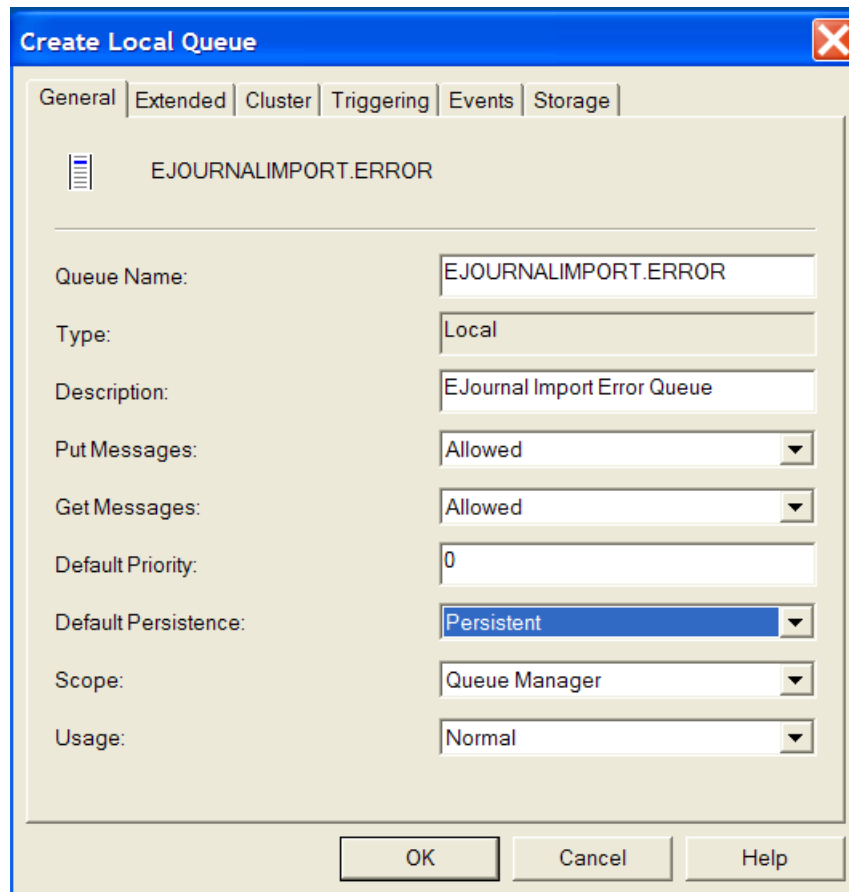
1. Expand the tree in the left navigation area to show `bo.queue.manager`.
2. Right-click the `Queues` folder under the `bo.queue.manager` (see Figure 3-1).

Figure 3-5 MQ Explorer: Queues Folder



3. Select **New, Local Queue**.
4. Enter the queue name in all capital letters (see Figure 3-2). For example, if you are creating an error queue for the EJournal Import queue, enter `EJOURNALIMPORT.ERROR`.
5. Delete the existing default description.
6. Enter a queue description, for example, `EJournal Import Error Queue`.
7. For Default Persistence, use the drop-down menu to choose **Persistent**.
8. Accept all other defaults.

Figure 3-6 Create Local Queue Window: Error Queue



The 'Create Local Queue' dialog box is shown with the 'General' tab selected. The queue name is 'EJOURNALIMPORT.ERROR'. The type is 'Local'. The description is 'EJournal Import Error Queue'. The 'Put Messages' and 'Get Messages' options are both set to 'Allowed'. The default priority is '0'. The default persistence is 'Persistent'. The scope is 'Queue Manager' and the usage is 'Normal'. The dialog has 'OK', 'Cancel', and 'Help' buttons at the bottom.

Property	Value
Queue Name:	EJOURNALIMPORT.ERROR
Type:	Local
Description:	EJournal Import Error Queue
Put Messages:	Allowed
Get Messages:	Allowed
Default Priority:	0
Default Persistence:	Persistent
Scope:	Queue Manager
Usage:	Normal

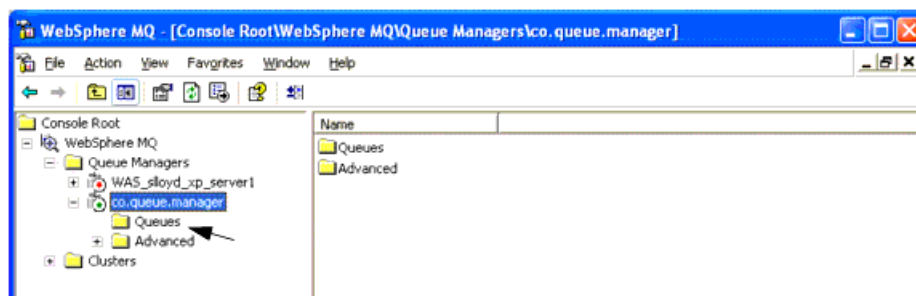
9. Click **OK**.

Creating a Local Queue using MQ Explorer

To create a local queue:.

1. Expand the tree in the left navigation area to show `bo.queue.manager`.
2. Right-click the **Queues** folder under the `bo.queue.manager` (see Figure 3-3).

Figure 3-7 MQ Explorer: Queues Folder



3. Select **New, Local Queue**.
4. Enter the queue name in all capital letters (see Figure 3-4). For example, `EJOURNALIMPORT`.

5. Delete the existing default description.
6. Enter a queue description, for example, EJournal Import Queue.
7. For Default Persistence, use the drop-down menu to choose **Persistent**.
8. Accept all other defaults.

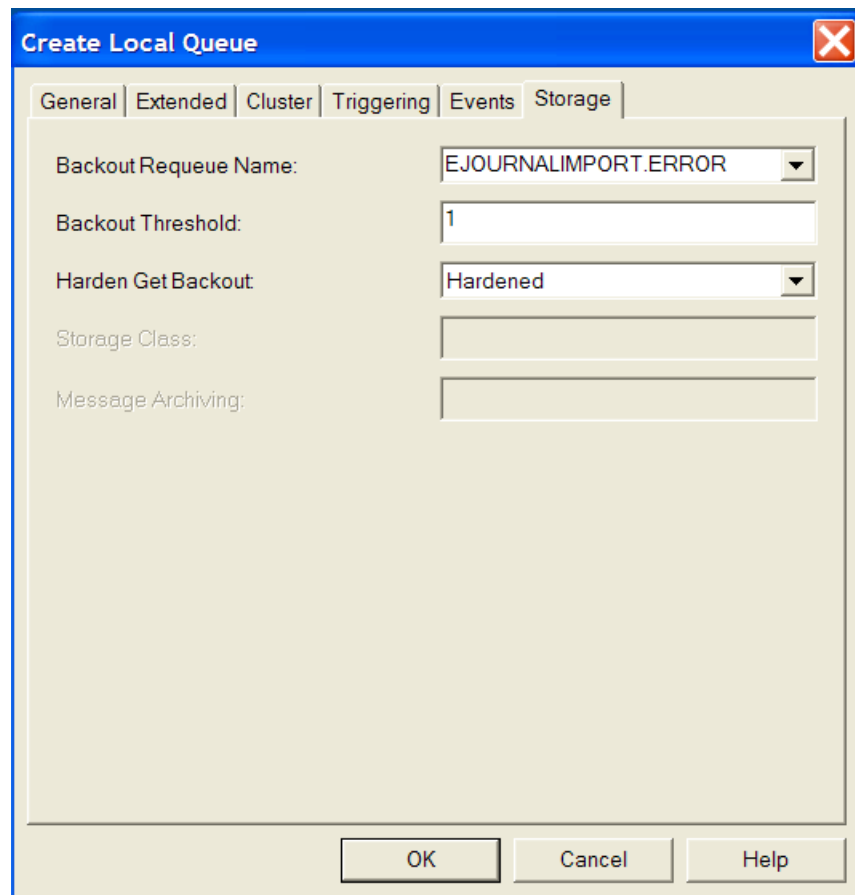
Figure 3-8 Create Local Queue Window

The screenshot shows the 'Create Local Queue' dialog box with the following configuration:

Property	Value
Queue Name	EJOURNALIMPORT
Type	Local
Description	EJournal Import queue
Put Messages	Allowed
Get Messages	Allowed
Default Priority	0
Default Persistence	Persistent
Scope	Queue Manager
Usage	Normal

9. If the queue has an error queue, configure it to use that error queue.
10. Choose the Storage tab.
11. Select the appropriate error queue in the Backout Requeue Name drop-down list. For example, for the EJOURNALIMPORT queue, select the EJOURNALIMPORT.ERROR queue. See Figure 3-5.
12. Change the backout threshold to 1. This enables MQ to place messages onto the backout queue (the error queue) after they have been processed once and rolled back.

Figure 3-9 Create Local Queue: Enter Backout Requeue Name

The image shows a 'Create Local Queue' dialog box with a blue title bar and a close button (X) in the top right corner. The dialog has several tabs: 'General', 'Extended', 'Cluster', 'Triggering', 'Events', and 'Storage'. The 'General' tab is selected. Inside the 'General' tab, there are five fields: 'Backout Requeue Name:' with a dropdown menu showing 'EJOURNALIMPORT.ERROR'; 'Backout Threshold:' with a text box containing '1'; 'Harden Get Backout:' with a dropdown menu showing 'Hardened'; 'Storage Class:' with an empty text box; and 'Message Archiving:' with an empty text box. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

13. Click **OK**.

Creating Multiple Queues Using an Input Data File

Use the following procedure to create multiple queues at once using an input data file. For more information about this file and an alternative procedure, see the WebSphere MQ documentation.

To create several queues at once using an input data file:

1. Create a text file with commands to create each queue. The following example is named `createq.dat`:

```
DEFINE QLOCAL (EJ.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (EJOURNALIMPORT) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(EJ.ERROR)
BOTHRESH(1)
DEFINE QLOCAL (POSLOG.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (POSLOG) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(POSLOG.ERROR) BOTHRESH(1)
DEFINE QLOCAL (SIGCAP.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (SIGNATURECAPTURE) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(SIGCAP.ERROR)
BOTHRESH(1)
DEFINE QLOCAL (TASKEXECUTION.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (TASKEXECUTION) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(TASKEXECUTION.ERROR)
```



```

BOTHRESH(1)
DEFINE QLOCAL (TAXIMPORT.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (TAXIMPORT) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(TAXIMPORT.ERROR)
BOTHRESH(1)
DEFINE QLOCAL (ITEMIMPORT.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (ITEMIMPORT) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(ITEMIMPORT.ERROR)
BOTHRESH(1)
DEFINE QLOCAL (PRICINGIMPORT.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (PRICINGIMPORT) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(PRICINGIMPORT.ERROR)
BOTHRESH(1)
DEFINE QLOCAL (AUDITLOGEXPORT.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (AUDITLOGEXPORT) USAGE (NORMAL) DEFPSIST(YES)
BOQNAME(AUDITLOGEXPORT.ERROR) BOTHRESH(1)
DEFINE QLOCAL (STORE_04241.ERROR) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_04241) USAGE (NORMAL) DEFPSIST(YES) BOQNAME(STORE_04241.ERROR)
BOTHRESH(1)
DEFINE QLOCAL (PARAMETERIMPORT) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (ACKNOWLEDGEMENT) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_04242) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_01291) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_00001) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_00002) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_00003) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_00004) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_00005) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_44444) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_66666) USAGE (NORMAL) DEFPSIST(YES)
DEFINE QLOCAL (STORE_77777) USAGE (NORMAL) DEFPSIST(YES)

```

Note: Error queues must be created before the normal queues which reference them.

2. Execute runmqsc with the text file as input, as shown below:

```
runmqsc bo.queue.manager < /root/install/<build number>/backofficeInstall/appservers/
was/createq.dat
```

3. Configure MQ Manager to use topics by running the following command:

```
runmqsc bo.queue.manager < /opt/mqm/java/bin/MQJMS_PSQ.mqsc
```

WebSphere Resources Required by Back Office

Before you begin configuration, WebSphere MQ and WebSphere Application Server must be installed.

Refer to the *IBM WebSphere Application Server V5.1 System Management and Configuration WebSphere Handbook* for complete information about deploying an application and managing WebSphere. This, as well as other detailed documentation, is located at <http://www.redbooks.ibm.com/>. You can also find extensive information about WebSphere MQ on this site.

The majority of the configuration you must complete prior to installing and deploying Back Office is done through the WebSphere Administration Panel. As the WebSphere administrator, your job is to define the resources required by Back Office. These must be configured in WebSphere prior to installing Back Office.

The following WebSphere resources are used by Back Office:

- Java Database Controller (JDC) resources—After you register a data source object with a Java Naming and Directory Interface (JNDI) naming service, an application can retrieve it from the naming service and use it to make a connection to the data source it represents.
- Mail Provider (JavaMail)—Configure a mail provider to enable e-mail alerts for approvals and task execution status. The default reference is `mail/Mail`.
- J2EE Connector Architectural Specification (JCA) Resources—As the WebSphere administrator, you must do the following:
 - Install and define the resource adapter.
 - Define one or more connection factories associated with the resource adapter.
- URL Providers—A Uniform Resource Locator (URL) is an identifier that points to a resource such as a directory on a network machine, a file in a directory, or a document stored in a database.
- Java Mail Session (JMS) Providers—A JavaMail session object is used by the application to obtain connections to a mail server. The default is `mail/Mail`.

Other Requirements

In addition, the following must be installed and/or configured:

- Java Development Kit (JDK)—WebSphere comes with the IBM JDK version 1.4.2. To support headless operation, add the following property to your Java command line:

```
-Djava.awt.headless=true
```

Note: If you use headless operation, X11 Virtual Frame Buffer (XVFB) must be installed and running, and the `DISPLAY` global variable must be set properly for the Back Office WebSphere Application Server. For more information, see the SUSE LINUX documentation.

- Quartz Scheduler—Back Office uses Quartz, an open-source product, to schedule jobs. For more information about Quartz, see the Quartz documentation at <http://www.quartzscheduler.org/quartz/>. (Quartz is packaged in the `.ear` file.) This release of Back Office uses Quartz version 1.3. Edit the `quartz.properties` file data sources to match the data sources you are using with Back Office and to make Quartz store its data in a database rather than in memory. If you do not make any changes to the data sources, you do not have to modify `quartz.properties`. In the following code sample, two lines define the two standard data sources:

```
org.quartz.dataSource.DEFAULT.jndiURL = java:/jdbc/DataSource  
org.quartz.dataSource.ALT.jndiURL = java:/jdbc/Other
```

- Logging—Back Office uses a Log4J component for logging. You can configure various elements of the Log4J operation. See the documentation for your application server for information about configuring logging in your environment.

Note: Commonly, you must configure the file path of the Back Office log. This requires changing the value of `log4j.appender.R` in the `log4j.properties` file.

Finally, you must increase the default Java Virtual Machine (JVM) heap size for your JDK to an appropriate level for your server and resource needs. Again, refer to your server documentation for specific information.

This book provides an introduction to these services as they apply directly to Back Office. For the most detailed and up-to-date information, refer to your WebSphere documentation or the IBM Redbooks (<http://www.redbooks.ibm.com/>).

Back Office is configured separately. For information about configuring Back Office, see Chapter 5, “Configuring Back Office”

Configuring WebSphere Application Server (WAS)

The Back Office application is packaged as a J2EE enterprise application archive (an `.ear` file), which relies on access to external resources using the application server. “Copying Files” on page 3-35 describes how to acquire the necessary files, including the `.ear` file. After performing the procedures described in that section, the `.ear` file is located in the `/install/<build number>/backofficeInstall` directory.

The following sections describe the two ways to configure WebSphere:

- “Configuration Using Scripts”
- “Manual Configuration”

If you use 360Commerce scripts for your configuration, two things must still be configured manually: security and server variables.

In addition, you need to copy the necessary files. See “Copying Files” on page 3-35.

Configuration Using Scripts

Use the following procedures to configure WebSphere using scripts. Four procedures are required:

- “Configuring the WAS Properties”
- “Building the Security Jar”
- “Configuring WebSphere and Back Office Security”
- “Configuring Additional Server Configuration for Back Office”

Configuring the WAS Properties

In the following procedure, `node.name` must be the machine name and is case-sensitive. Use the `hostname` command to see the machine name. In this procedures, you must change the following values:

- `db.password`
- `db.jdbc-url`
- `ear.file`
- `node.name`
- `mq.queue.manager`
- `jvmArgs`

To configure the WAS properties:

1. Execute the following command:

```
cd /install/<build number>/backofficeInstall/backoffice/appservers/was
```

2. Edit the `was.properties` file to conform to the following example:

```
db.user=db2inst1
db.password=db2pwd
```

```

db.jdbc-url=jdbc:db2://10.100.19.2:50001/quarrysb
temp.dir=/tmp
was.home=/opt/WebSphere/AppServer
was.libs=${was.home}/lib
task.lib=${was.libs}/wsantasks.jar
app.name=BackOffice
ear.file=/backoffice/backoffice.ear
server.name=server1
node.name=IresQA
server.user=pos
server.passwd=pos
#server.user=uid=wasadmin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot
#server.passwd=wasadmin
server.connection=SOAP
mq.user=mqm
mq.passwd=mqm
mq.queue.manager=bo.queue.manager
data.source=jdbc/DataSource
registry.class=com._360commerce.commerceservices.security.websphere.COUserRegistry
jvmArgs=-Djava.awt.headless=true -Duser.timezone=America/Chicago
db.install.path=${was.libs}

```

Building the Security Jar

To build the security jar:

1. Execute the following command:

```
cd /opt/WebSphere/AppServer/bin
```

2. Execute the following command:

```
./ws_ant -f /backoffice/appservers/was/build.xml build_security_jar
```

Configuring WebSphere and Back Office Security

Configuring WebSphere and Back Office security must be done manually. See “Security” on page 3-32. Perform the procedures in the following order:

1. Set custom properties.
2. Configure user registries.
3. Configure global security.

Configuring Additional Server Configuration for Back Office

To configure additional server configurations for Back Office:

1. Execute the following command:

```
cd ~/install/<build number>/backofficeInstall
```

2. Execute the following series of commands:

```
ws_ant configure
```

```
ws_ant installApp
```

```
ws_ant startApp
```

Manual Configuration

Start the WebSphere Application Server as you normally do. Log into the WebSphere Application Server using a browser. The Administration Panel is defined into a left navigation area and a content pane. The following topics include the procedures to configure each aspect of the WebSphere Application Server:

- “Application Servers”
- “Resources”

Application Servers

Configuring application servers includes three procedures:

- “Setting Java Virtual Machine (JVM) Options”
- “Setting Custom Properties”
- “Configuring the Listener Services”

Setting Java Virtual Machine (JVM) Options

The IBM default time zone for the JVM differs from the Sun default; therefore, you may want to set this value. 360Commerce also recommends that you increase the default heap size.

To set JVM options:

1. In the left navigation area of the WAS Administrative Panel, select **Servers**.
2. Select **Application Servers**.
3. Select the server, for example, **server1**.
4. Scroll down and select **Process Definition**.
5. In the Additional Properties section, choose **Java Virtual Machine**.
6. Under Generic JVM Arguments, enter `-Duser.timezone=America/Chicago` or an appropriate setting for your location.
7. Scroll to the Initial Heap Size field and set an appropriate value that is greater than the default value of 0.

8. Click **OK**.

Setting Custom Properties

To set custom properties:

1. In the left navigation area of the WAS Administrative Panel, select **Servers**.
2. Select **Application Servers**.
3. Select the server, for example, server1.
4. Scroll down and select **Process Definition**.
5. In the Additional Properties section, choose **Environment Entries**.
6. For the Environment Entries, enter the values listed in the following table:

Table 3-2 Server Custom Properties

Name	Value	Definition
DB2INSTANCE	db2inst1	DB2 Environmental Variable
DISPLAY	:1.0	Environmental variable for xvfb
LANG	en_US	
LD_ASSUME_KERNEL	2.4.19	Kernel Threading Type Variable
LD_LIBRARY_PATH	/home/db2inst1/sqllib/lib	LD_LIBRARY Environmental Variable
LIBPATH	/home/db2inst1/sqllib/lib	LIBPATH Environmental Variable

Configuring the Listener Services

To configure the listener services:

1. In the left navigation area of the WAS Administrative Panel, select **Servers**.
2. Select **Application Servers**.
3. Select the server, for example, server1.
4. Scroll to the Additional Properties section.
5. Select **Message Listener Service**.
6. Select **Listener Ports**.
7. Click **New**.
8. Enter the necessary values. The following table provides an example.

Table 3-3 Example: Message Listener Service Configuration

Field	Value
Name	EJournalImportPort
InitialState	Started
Description	(optional)
Connection factory JNDI name	jms/QueueConnectionFactory
Destination JNDI name	jms/EJournalImport
Maximum Sessions	5
Maximum Retries	0
Maximum Messages	100

1. Click **OK**.
2. Repeat steps 7-1 to add all required listener ports. See the following table for a list of required listener ports.

Table 3-4 Listener Ports

Name	JNDI Connection Factory	JNDI Name
AcknowledgementPort	jms/QueueConnectionFactory	jms/Acknowledgement
EJournalImportPort	jms/QueueConnectionFactory	jms/EJournalImport
ItemImportPort	jms/QueueConnectionFactory	jms/ItemImport
POSLogPort	jms/QueueConnectionFactory	jms/POSLog
PricingImportPort	jms/QueueConnectionFactory	jms/PricingImport
AuditLogExportPort	jms/QueueConnectionFactory	jms/AuditLogExport
ParameterImportPort	jms/QueueConnectionFactory	jms/ParameterImport
SignatureCapturePort	jms/QueueConnectionFactory	jms/SignatureCapture
Store XXXXXMDBPort*	jms/QueueConnectionFactory	jms/store_XXXXX*
TaskExecutionPort	jms/QueueConnectionFactory	jms/TaskExecution
TaxImportPort	jms/QueueConnectionFactory	jms/TaxImport
* Each store must have a queue defined; XXXXX represents the store number. Use the format shown, as Back Office expects this nomenclature.		

Resources

You must perform the following procedure to configure the necessary resources:

- “Configuring the JDBC XA Provider”
- “Configuring the JDBC Non-XA Provider”
- “Configuring the Queue Connection Factory”
- “Defining the MQ Queue Destinations”
- “Creating a Mail Session”

You must configure WebSphere Application Server with the destination for each queue.

Configuring the JDBC XA Provider

To configure the JDBC XA provider:

1. In the left navigation area of the WAS Administrative Panel, select **Resources**.
2. Select **JDBC Providers**.
3. Click **New**.
4. Use the drop down menu to select the type of JDBC Provider to be configured.
5. Click **OK**.
6. Enter the necessary values. The following table provides an example.

Table 3-5 Example: JDBC XA Provider

Field	Value
Name	DB2 JDBC Thin Driver (XA)
Description	DB2 JDBC Thin Driver (XA)
Classpath	\${DB2_JDBC_DRIVER_PATH}/db2jcc.jar
Implementation Classname	DB2jdbc.xa.client.DB2XADataSource

Note: Refer to the WebSphere documentation for how to set variables such as \${DB2_JDBC_DRIVER_PATH} in the WebSphere environment.

1. Click **OK**. The JDBC Providers page appears.
2. Choose the JDBC Provider just created, for example, DB2 JDBC Thin Driver (XA).
3. Scroll to the Additional Properties section.
4. Choose **Data Sources**.
5. Click **New**. The Data Sources New page appears.
6. Enter the necessary values See the following table for an example.

Table 3-6 Example: JDBC XA Data Source

Field	Value
Name	DB2 DataSource (XA)
JNDI Name	JDBC/DataSource
Container managed persistence	Unchecked
Description	(optional)
Category	(optional)
Statement Cache Size	10
Datasource Helper Classname	com.ibm.websphere.rsadapter.DB2DataStoreHelper
Component-managed Authentication Alias	<hostname>/DB2XAAuthAlias
Container-managed Authentication Alias	<hostname>/DB2XAAuthAlias

7. Click **OK**.

8. Choose the Data Source just entered, for example, DB2 DataSource (XA).
9. Scroll down to the Additional Properties section.
10. Choose **Custom Properties**.
11. Click the text URL.
12. Enter the value for the Database Installation address, for example, jdbc:DB2:thin:@<servername>:1521:quarrysb.
13. Click **OK**.
14. Choose the JDBC provider page again, for example, DB2 JDBC Thin Driver (XA).
15. Choose **Data Sources (Version 4)**.
16. Click **New** to enter an additional data source for Crystal Clear.
17. Enter the necessary values. See the following table for an example.

Table 3-7 Example: JDBC XA Crystal Clear Data Source

Field	Value
Name	CrystalClearDS
JNDI Name	jdbc/CrystalClearDS
Container managed persistence	Unchecked
Description	(optional)
Category	(optional)
Database Name	<database product name>
Default User ID	<database user ID>
Default Password	<database password>

18. Click **OK**. The Data Sources page appears.
19. Choose the data source just entered, for example, DB2 DataSource (XA).
20. Scroll down to the Additional Properties section.
21. Choose **Custom Properties**.
22. Add the following custom properties:
 - driverType = 4
 - portNumber = <port number of DB2 server>
23. Enter the value for the database installation address, for example, jdbc:DB2:thin:@<servername>:1521:quarrysb.
24. Click **OK**.
25. Return to the Data Sources page and click **Test Connection** to verify your settings.

Configuring the JDBC Non-XA Provider

To configure the JDBC Non-XA provider:

1. In the left navigation area of the WAS Administrative Panel, select **Resources**.

2. Select **JDBC Providers**. The JDBC Providers screen opens.
3. Click **New**.
4. Use the drop down menu to choose the type of JDBC Provider to be configured.
5. Click **OK**.
6. Enter the necessary values. See the following table for an example.

Table 3-8 Example: DB2 JDBC Thin Driver

Field	Value
Description	DB2 JDBC Thin Driver
Classpath	\${DB2_JDBC_DRIVER_PATH}/db2jcc.jar
Implementation Classname	DB2.jdbc.pool.DB2ConnectionPoolDataSource

1. Click **OK**. The JDBC Providers screen opens.
2. Choose the JDBC Provider just created, for example, DB2 Thin Driver.
3. Scroll down to the Additional Properties section.
4. Choose **Data Sources**. The Data Sources screen opens.
5. Choose **New**. The New screen opens.
6. Enter the necessary values. The following table provides an example.

Table 3-9 Example: JDBC (non-XA) DataSource Configuration

Field	Value
Name	DB2 DataSource
JNDI Name	jdbc/Other
Container managed persistence	Unchecked
Description	(optional)
Category	(optional)
Statement Cache Size	10
Datasource Helper Classname	com.ibm.websphere.rsadapter.DB2DataStoreHelper
Component-managed Authentication Alias	host name/DB2AuthAlias
Container-managed Authentication Alias	host name/DB2AuthAlias

7. Click **OK**. The Data Sources screen opens.
8. Choose the DataSource just entered, or example, DB2 DataSource.
9. Scroll down to the Additional Properties section.
10. Choose **Custom Properties**.
11. Click the text URL.
12. Enter the value for the database installation address, for example, jdbc:DB2:thin:@172.16.128.110:1521:quarrysb.
13. Click **OK**.

Configuring the Queue Connection Factory

To configure the Queue Connection Factory:

1. In the left navigation area of the WAS Administrative Panel, select **Resources**.
2. Choose **WebSphere MQ JMS Provider**.
3. In the content pane, scroll to the Additional Properties section.
4. Choose **WebSphere MQ Queue Connection Factories**.
5. Choose **New**.
6. For Name, enter `COQueueConnectionFactory`.
7. For JNDI Name, enter `jms/QueueConnectionFactory`.
8. For Component-managed Authentication Alias, select the host name.
9. For Container-managed Authentication Alias, select the host name.
10. For Queue Manager, enter `bo.queue.manager`.
11. For Host, enter the MQ host name.
12. For Port, enter `1414`.
13. For Channel, select **SYSTEM.DEF.SVRCONN**.
14. For Transport Type, enter `CLIENT`.
15. For Message Retention, check **Enable Message Retention**.
16. For XA Enabled, check **Enable XA**.
17. Click **OK**.

Defining the MQ Queue Destinations

To define MQ queue destinations:

1. In the left navigation area of the WAS Administrative Panel, select **Resources**.
2. Choose **WebSphere MQ JMS Provider**.
3. Scroll to the Additional Properties section in the content pane.
4. Choose **WebSphere MQ Queue Destinations**.
5. Click **New**.
6. Enter the data specific to the queue destination being configured. The following table illustrates how the fields should be configured using the EJournalImport queue as an example.

Table 3-10 Example: MQ Queue Destination Options

Field	Value
Name	EJournalImport
JNDI Name	jms/EJournalImport
Description	(optional)
Category	(optional)
Persistence	QUEUE DEFINED
Priority	QUEUE DEFINED
Expiry	SPECIFIED
Specified Expiry	60000
Base Queue Name	EJOURNALIMPORT
Base Queue Manager	bo.queue.manager
Native Encoding	Unchecked
Integer Encoding	Normal
Decimal Encoding	Normal
Floating Point Encoding	IEEENormal
Target Client	JMS
Queue Manager Host	MQ Host
Queue Manager Port	1414
Server Connection Channel Name	SYSTEM.DEF.SRVCONN
UserID	Windows: {leave blank} Unix: MQ userid
Password	Windows: {leave blank} Unix: MQ password

Note: The Base Queue name is the MQ queue name. Both names should be in all capital letters.

1. Click **OK**.
2. Repeat steps 5-1 for each queue. Enter a destination for all the queues entered when you configured WebSphere MQ. See the following table for a list of typical queues.

Table 3-11 Typical Queues

JNDI Name	Description
jms/EJournalImport	EJournal import queue
jms/POSLog	POSLog import queue
jms/ItemImport	Item import queue
jms/AuditLogExport	Audit log export queue
jms/TaxImport	Tax import queue
jms/PricingImport	Pricing import queue
jms/ParameterImport	Parameter import queue
jms/TaskExecution	Workflow scheduling/execution queue.
jms/SignatureCapture	Custom queue for importing signature captures assuming no POSLog is coming
jms/Acknowledgement	Acknowledgement that subscriber received message
jms/store_XXXXX	One for each store

Creating a Mail Session

To create a mail session:

1. In the left navigation area of the WAS Administrative Panel, select **Resources**.
2. Choose **Mail Providers**.
3. Choose **Built-in Mail Provider**.
4. In the Additional Properties section in the content pane, choose **Mail Sessions**.
5. Click **New**.
6. Enter the values specific to your deployment. The following table provides an example.

Table 3-12 Example: MQ Queue Destination Options

Field	Value
Name	BackOfficeMail
JNDI Name	mail/Mail
Mail Transport Host	<i>Set to an appropriate value for your deployment</i>

Security

Configuring security includes the following procedures:

- Configuring User Registries
- Configuring Global Security
- Configuring the JAAS Java 2 Connector Architecture Aliases

Configuring User Registries

To configure user registries:

1. In the left navigation area of the WAS Administrative Panel, select **Security**.
2. Choose **User Registries**.
3. Choose **Custom**.
4. Enter the necessary values. See the following table for an example.

Table 3-13 Example: User Registry Security Configuration

Field	Value
Server User ID	UserID
Server User Password	password
Custom Registry Classname	com._360commerce.commerceservices.security.websphere.COUserRegistry
Ignore Case	Checked

5. Click **Apply**.
6. Under Additional Properties, select **Custom Properties**.
7. Add the following custom properties:
 - jdbcURL — jdbc:db2://10.100.19.2:50001/quarrysb
 - jdbcDriver — com.ibm.db2.jcc.DB2Driver
 - jdbcUser — db2inst1
 - jdbcPass — db2pwd
8. Click **OK**.

Configuring Global Security

To configure global security:

1. In the left navigation area of the WAS Administrative Panel, select **Security**.
2. Choose **Global Security**.
3. Enter the necessary values. See the following table for an example.

Table 3-14 Example: Global Security Configuration

Field	Value
Enabled	Checked
Enforce Java 2 Security	Unchecked
Use Domain Qualified User IDs	Unchecked
Cache Timeout	600
Issue Permission Warning	Checked
Active Protocol	CSI and SAS
Active Authentication Mechanism	SWAM
Active User Registry	Custom

1. Click **OK**.
2. Save the server settings.
3. Restart WebSphere. See “Restarting the WebSphere Application Server” on page 3-36.

Configuring the JAAS Java 2 Connector Architecture Aliases

To configure the JAAS Java 2 Connector Architecture (J2C) aliases:

1. In the left navigation area of the WAS Administrative Panel, select **Security**.
2. Choose **JAAS Configuration**.
3. Choose **J2C Authentication Data**.
4. In the content pane, choose **New**.
5. Enter the alias.
6. Enter the user ID.
7. Enter the password.
8. Enter a description.
9. Click **OK**.

Table 3-15 Example: J2C Authentication Data for XA Resource

Field	Value
Alias	DB2 XAAuthAlias
UserID	DB2 ID for the XA resource
Password	DB2 password for the XA resource
Description	(optional)

10. Click **New**.
11. Enter the alias name.
12. Enter the user ID.

13. Enter the password.
14. Enter a description.
15. Click **OK**.

Table 3-16 Example: J2C Authentication Data for non-XA resource

Field	Value
Alias	DB2 AuthAlias
UserID	DB2 ID for the non-XA resource
Password	DB2 password for the non-XA resource
Description	(optional)

Copying Files

Go to the `/install/<build number>/backofficeInstall` directory and copy files from there to your WebSphere installation folder. See the following table for an example that uses DB2 and the 360Commerce custom user registry.

Table 3-17 Example: Files to Copy

File	From Path (relative to <code><prod_install_dir></code>)	To Path (Relative to Application Server install folder)	Description
security360.jar	\appservers\was	\lib\ext	Custom user registry for use with 360Commerce security; not needed if using LDAP security.
log4jxxx.jar			Jakarta logging utility, where xxx represents the version.
Database driver .jar or .zip file	Get appropriate driver version from your database vendor		For example, db2jcc.jar or db2java.zip for DB2.
quartz.properties	\appservers\was	\properties	Quartz configuration.
log4j.properties			Log4J configuration.
integral-jms-authorizations.xml		\config\cells\ <nodename> example: \config\cells\server1	Back Office authorizations. Place a copy in each node's folder in \cells. See the IBM WebSphere Application Server documentation for further information.

Deploying Back Office

Deploying Back Office includes two steps:

- “Installing Back Office”
- “Restarting the WebSphere Application Server”

Installing Back Office

To install the Back Office application:

1. In the left navigation area of the WAS Administrative Panel, select **Applications**.
2. Choose **Install New Application**.
3. In the content pane, specify the path to the `backoffice.ear` file.
4. Choose **Next**. The Preparing for Application Installation screen opens.
5. Choose **Next**. The Install New Application screen opens. This screen provides a list of hyperlinked steps for configuring an installed application. The configuration tasks handled by those links are covered by other procedures in this chapter or by default data included with the `backoffice.ear` file.
6. Choose the final step on the list (Step 13).
7. Choose **Finish**. The Administrative Console displays a stream of messages as it installs the application. If it completes successfully, it displays a success message.
8. Scroll to the end of the list of messages and choose **Save to Master Configuration**. The Save screen opens.
9. Choose **Save**.

Restarting the WebSphere Application Server

Restart the application server to enable security.

To restart the server with new variables:

1. At the prompt, create a new startup script using the following commands:

```
cd /opt/WebSphere/AppServer/bin
./startServer.sh server1 -script
```

2. To solve a current IBM bug, edit the `start_server1.sh` script:

- a. Run the following command:

```
vi start_server.sh
```

- a. Comment out the `exec` statement.
- b. Add below it a statement like the following:

```
./startServer.sh server1
```

3. Edit the `j2c.properties` file to change the `cm-properties` tag element:

```
<!-- The cm-properties are in a comment block. Uncomment to use -->
<cm-properties>
    <!--<manageCachedHandles>false</manageCachedHandles>-->
    <logMissingTranContext>false</logMissingTranContext>
</cm-properties>
```

4. Stop and restart the server with the following commands:

```
cd /opt/WebSphere/AppServer/bin/
stopServer.sh
start_server.sh &
```


CONFIGURING THE ACCESSVIA PRINT ENGINE FOR LABELS AND TAGS

In order to use the Labels and Tags functionality of Back Office, you need to install and configure the AccessVia Print engine.

Before configuring the AccessVia Print engine, you must have completed the following procedures:

- The installation and configuration of all prerequisite software.
- The setup of the database.
- The deployment of the `backoffice.ear` file to the application server.
- The installation of the printer.

Configuring the AccessVia Print engine includes the following tasks:

- Performing the necessary configuration of the database. See “Configuring the Database for the AccessVia Print Engine.”
- Setting up the required directory structure. See “Setting Up the Directory Structure.”
- Creating the AccessVia `.ini` file. See “Creating the AccessVia Print Engine `.ini` File.”
- Setting the database pointer to the `.ini` file. See “Setting the Database Pointer.”
- Setting the necessary variables. See “Setting Variables for the AccessVia Print Engine.”
- Testing the AccessVia Print engine. See “Testing Standalone Printing.”

In addition, see XX to troubleshoot printing errors.

Configuring the Database for the AccessVia Print Engine

Because Labels and Tags needs to access data from Back Office, AccessVia requires open database connectivity (ODBC) to the Back Office database. AccessVia stores template information in the following Back Office data tables:

- **SGFORM**—This table stores templates.
- **SGELEM**—This table stores template attributes.
- **SGSQL**—This table stores .zip files of SQL, which fetch template data at the time of printing.
- **SGCONFIG**—This table stores the paths for .ini files required by AccessVia.

Configuring DB2

The DB2 client software must be installed and configured and the database must be catalogued. DB2 CLI must be enabled.

Configuring DB2 for the AccessVia Print engine includes 5 tasks:

- Creating an `.odbc.ini` file.
- Modifying the `db2cli.ini` file.
- Setting up the DB2 user profile.
- Enabling the user to connect to DB2 as a client.
- Testing the configuration.

To create an `.odbc.ini` file:

1. Locate the home directory of the DB2 user, as in the following example:

```
/home/<db2 user>
```

2. Create the `.odbc.ini` file within that directory, as in the following example:

```
/home/<db2 user>/odbc.ini
```

3. Add the following lines to the `.odbc.ini` file:

```
[ODBC Data Source]
```

```
<db2 instance>=IBM DB2 ODBC DRIVER
```

```
[<db2 instance>]
```

```
Driver=/home/<db2 user>/sqllib/lib/libdb2.so
```

```
Description=<DB2 ODBC Database>
```

To modify the db2cli.ini file:

1. Locate /home/<db2 user>/sqllib/cfg/db2cli.ini.
2. Add the following lines to the db2cli.ini file:

```
[SLM]

uid=db2inst1

pwd=<password for db2 user>

autocommit=0

TableType="'TABLE', 'VIEW', 'SYSTEM TABLE'"

DBALIAS=<db2 instance>

DESCRIPTION=SLM
```

To set up the DB2 user profile:

1. Locate /home/<db2 user>/profile.
2. Add the following lines to the .profile file:

```
#[IBM DB2 ODBC DRIVER]

Driver=/home/<db2 user>/sqllib/lib/libdb2.so

DB2INSTANCE=<db2 user>

export ODBCINI=/home/<db2 user>/odbc.ini
```

To enable the user to connect to DB2 as a client:

Run the following command:

```
./home/<db2 user>/sqllib/db2profile
```

To test the configuration:

At the command prompt, enter db2. If you can get to the DB2 command prompt, DB2 is properly configured.

Configuring MySQL

Configuring MySQL for the AccessVia Print engine requires establishing an ODBC connection.

To establish a MySQL ODBC connection:

1. Copy the ODBC .exe file from thirdparty\accessvia\windows\odbcdriver.
2. Install the ODBC driver.
3. In Windows, select **Start > Control Panel > Administrative Tools > Data Sources(ODBC)**.
4. Select the User DSN tab.
5. Click **Add**. The Create New Data Source box opens.

6. Select which driver you want to set up a data source.
7. Click **Finish**. The Connector/ODBC for MySQL box opens.
8. Complete the configuration by entering the login information.
9. Test the connection.

Setting Up the Directory Structure

Both operating systems require the same directories.

Setting up the Linux Directory Structure

Set up the following directory structure for the AccessVia Print engine:

- /opt/accessvia/
 - *dst*—This is the *user* directory. It should include subdirectories for data, images, and fonts. The *FormatPath*, *WorkPath*, and *Userpath* for the [STARTUP] section of the .ini file lead to this directory.
 - *data*—The *DataPath* for the [STARTUP] section leads to this directory. Put all necessary data here.
 - *images*—The *GraphicPath* for the [STARTUP] section leads to this directory. Put all necessary graphic images here.
 - *fonts*—The *FontPath* for the [STARTUP] section leads to this directory. Put all necessary fonts here.
 - *program*—The program directory must include all the AccessVia Print engine libraries. These can be copied from the 360Commerce `thirdparty\accessvia\linux\suse\8.1\program` directory. The *ExePath* for the [STARTUP] section leads to this directory.
 - *system*—The system directory must include all system files. These can be copied from the 360Commerce `thirdparty\accessvia\linux\suse\8.1\system` directory. The *SystemPath* for the [STARTUP] section leads to this directory.

Setting up the Windows Directory Structure

Set up the following directory structure for the AccessVia Print engine:

- C:\accessvia\
 - *dst*—This is the *user* directory. It should include subdirectories for data, images, and fonts. The *FormatPath*, *WorkPath*, and *Userpath* for the [STARTUP] section of the .ini file lead to this directory.
 - *data*—The *DataPath* for the [STARTUP] section leads to this directory. Put all necessary data here.
 - *images*—The *GraphicPath* for the [STARTUP] section leads to this directory. Put all necessary graphic images here.

- **fonts**—The *FontPath* for the [STARTUP] section leads to this directory. Put all necessary fonts here.
- **program**—The program directory must include all the AccessVia Print engine libraries. These can be copied from the 360Commerce `thirdparty\accessvia\windows\program` directory. The *ExePath* for the [STARTUP] section leads to this directory.
- **system**—The system directory must include all system files. These can be copied from the 360Commerce `thirdparty\accessvia\windows\system` directory. The *SystemPath* for the [STARTUP] section leads to this directory.

Creating the AccessVia Print Engine .ini File

The AccessVia Print engine requires an .ini file for configuration. This file controls all AccessVia operations.

The default name for the AccessVia .ini file is `dsign.ini`, and that name is used to refer to it throughout this chapter.

.ini File Settings

This file contains a series of settings:

- **Path settings**—These are used by the AccessVia APIs to fetch appropriate attributes at the time of printing. These paths, which are located in the `System Setup` section, lead to the directories described in “Configuring MySQL.” *GraphicPath*, *FontPath*, and *ExePath* must point to individual folders; the remaining paths can point to a common folder because they are not used as often. In order for *UserPath* to be functional, Back Office must have write permission to the `dst` directory.
 - **DataPath**—This must point to the folder that contains all the necessary data (`data`).
 - **GraphicPath**—This must point to the folder that contains all images required for the print templates (`images`).
 - **FontPath**—This must point to the folder that contains all the font files required by the print templates (`fonts`).
 - **UserPath**—This must point to the user directory (`dst`).
 - **ExePath**—This must point to the folder that contains all AccessVia .dll files (`program`).
 - **SystemPath**—This must point to the folder that contains all necessary system files (`system`).
 - **WorkPath**—This must point to the folder used by AccessVia APIs to write temp files during the printing process.
- **Printer settings**—These are the printer attributes. They are located in the `Printer Setup` section. Most of them are the same as the system printer settings. *PrintFile*, *PrintToFile*, and *PrinterName* are the most important attributes; the remaining ones can use default settings.
 - `PrinterPort=WS:`
 - `PrintFile=<AccessVia root file>\temp\output.prn`

- PrintToFile= No. However, for initial testing, you can arrange for templates to be printed in an output file (PrintFile) by setting PrintToFile to Yes.
- PrinterDriver=POSTSCRIPT. The AccessVia Print engine prefers PostScript printers to PCL printers.
- PrinterName=Lexmark Optra T (or the default printer)
- PortSetting1=172.16.34.12. This printer IP address has proven successful for 360Commerce network printers.
- PortSetting2=9100. This port has proven successful for 360Commerce network printers.
- Data source settings—These provide AccessVia APIs with the location of templates and template data. These can be stored in the same place, in which case the two settings are identical. In the data sources, set the DSN name, database name, server name, user ID, and password correctly.
 - DATABASE—This is the data source for template data.
 - FORMATS—This is the data source for templates and template attributes.

.ini File Prototype

Create your .ini file to resemble the available prototype. The Linux and Windows prototypes are similar.

- Access the Linux prototype at `thirdparty\accessvia\linux\suse\8.1\program`.
- Access the Windows prototype at `thirdparty\accessvia\windows\program`.

```

;-----
;--- Database Connection Section -----
;-----

[DCM Global]

DataDriver=ODBC

ConnectRetry=4


;----- DATABASE Connection Properties -----

[DATABASE]

Enabled=True

DataDriver=ODBC

CONNECTION=DSN=SLM;DBALIAS=QUARRYSB

SCHEMA_SYS=<db2 user>


[SYSTEM]

Enabled=False

```

```

[FORMATS]

Enabled=False

;DataDriver=ODBC

;CONNECTION=DSN=quarrysb;DBALIAS=quarrysb

;CONNECTION=DSN=dsign;DBALIAS=DSIGN

;SCHEMA_SYS=<db2 user>


[IMPORTS]

Enabled=False


[EXPORTS]

Enabled=False


[STARTUP]

InitApp=No

;----- System Setup

DataPath=/opt/accessvia/dst/data

GraphicPath=/opt/accessvia/dst/images

FormatPath=/opt/accessvia/dst/

ExePath=/opt/accessvia/program/

SystemPath=/opt/accessvia/system/

FontPath=/opt/accessvia/dst/fonts/

WorkPath=/opt/accessvia/dst/

UserPath=/opt/accessvia/dst/


;----- Printer Setup

PrinterDriver=PS

;PrinterDriver=PM

;PrinterDriver=GDI

PrinterName=Lexmark Optra S

;PrinterName=\\360printserv\COPYWEST-RM127,WinPrint,IP_172.16.34.12

;PrinterName=Generic PS

PrinterPort=WS:

;PrintToFile=Yes

```

```
PrintToFile=No
PrintFile=output.ps
PrintSpooler=
BumpPageX=0
BumpPageY=0
PaperTray=
PrintCopies=1
PrintMode=No
SignOffset=1
PrinterPortMode=NEW
PrinterOptimizationType=NONE
PageTotal=No
PortSetting1=172.16.34.12
PortSetting2=9100
PortSetting3=9600,n,8,1

;----- Messaging and Errors
ErrorLog=design.err
;Debug=No
;MessageMode=SILENT
;DebugMode=SILENT
Debug=Yes
MessageMode=EXTENSIVE
DebugMode=EXTENSIVE

[ FONTS ]
```

Setting the Database Pointer

The application server JVM requires a pointer to the `.ini` file.

Setting the Database Pointer for WebSphere

WebSphere looks for the database pointer in the `sgconfig` table. If it fails to find it, it looks instead into its current directory, `/opt/WebSphere/Appserver/bin`.

To set the database pointer:

1. Locate the `sgconfig` table.
2. For WebSphere, make certain the `FCONFIGPARAMVALUE` column in the `sgconfig` table is set to `/opt/accessvia/program/dsign.ini` or the location of the AccessVia `.ini` file, if it differs from that.
3. Set the parameter `AccessViaIniFilePath = /opt/accessvia/program/dsign.ini`.

Setting the Database Pointer for JBoss

Copy the `.ini` file with the configuration information you created in “Creating the AccessVia Print Engine `.ini` File” to the current JBoss directory, `c:\jboss-4.0.2/bin`.

Setting Variables for the AccessVia Print Engine

Variables need to be set for the AccessVia Print engine to be functional.

Setting Variables for WebSphere

The `.jar` file that contains the AccessVia print engine API (`dJava.jar`) must be in the application server classpath, `/opt/WebSphere/Appserver/lib`.

`dJava.jar` can be copied from the 360Commerce `thirdparty\accessvia\linux\suse\8.1` directory.

The `LD_LIBRARY_PATH` variable must point to the database library path and the AccessVia Print engine `ExePath`.

To set the `LD_LIBRARY_PATH` variable:

Run the following command:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/IBM/db2/V8.1/lib:/opt/accessvia/program
export LD_LIBRARY_PATH
```

Setting Variables for JBoss

The `.jar` file that contains the AccessVia print engine API (`dJava.jar` file) must be in the application server classpath, `c:\jboss/server/default/lib`.

This file can be copied from the 360Commerce `thirdparty\accessvia` directory.

The `Path` environment variable must point to the location of the AccessVia Print engine.

To modify the Path variable:

From the command prompt, use the following command:

```
Set Path = %Path%;c:\accessvia\program
```

OR

1. In Windows, select **Start > Control Panel > System**. The System Properties box opens.
2. Select the Advanced tab.
3. Click **Environment Variables**.
4. Edit the `Path` variable to append the AccessVia Print engine location.

Testing Standalone Printing

After all of the previous steps have been completed, test the AccessVia Print engine.

Testing AccessVia in WebSphere

To test AccessVia in WebSphere:

1. Copy the test directory from `thirdparty\accessvia\linux\suse\8.1\test` to the `\opt\accessvia\test` directory.
2. Change the `.ini` file, located at `\accessvia\linux\suse\8.1\test`, to point to the correct data sources and to have the correct printer attributes.
3. Execute `test.sh`. Any errors that occur during the test are logged in `dsign.err` at `/opt/accessvia/dst/data`.

Testing AccessVia in JBoss

To test AccessVia in JBoss:

1. From the Windows command prompt, navigate to `thirdparty/accessvia/test`.
2. Locate the AccessVia `.ini` file.
3. Make the necessary changes in the `.ini` file to provide the correct database and printer information.

4. Edit the `runTest.bat` file to append the location of AccessVia to the `Path` variable, for example,

```
PATH = %PATH%;\ACCESSVIA\PROGRAM
```

5. Execute `runTest.bat`. Any errors that occur during the test are logged in `dsign.err` at `/thirdparty/accessvia/test/data`.

Troubleshooting

In the event of a failed attempt to print, one of the following error messages may appear:

- “MalformedURLException”
- “ClassNotFoundException”

MalformedURLException

The templates required by AccessVia are included in a comma-separated `.csv` file. If that file fails to import, printing cannot occur and a `MalformedURLException` occurs.

To correct a `MalformedURLException` error:

1. Determine whether the application is using UTF-8 encoding by examining the environment variable that specifies locale (`LANG` or `LC_ALL`) to see if it ends with `.UTF-8` (for example, `en_US.UTF-8`).
2. Remove the `.UTF-8` suffix and set `LANG` to `en_US`.

ClassNotFoundException

This is a class path error indicating that the `dJava.jar` file has not been included in the class path.

To correct a `ClassNotFoundException` error:

1. Make certain `dJava.jar` has been placed in the `lib` directory.
2. Edit `startServer.sh` (in the WebSphere `bin` directory) to append `-classpath`. The `startServer.sh` file should include the following line:

```
-classpath "$WAS_CLASSPATH:/opt/WebSphere/AppServer/lib\"
```


CONFIGURING BACK OFFICE

This chapter discusses the configuration steps that must be performed after Back Office is deployed.

- “Loading Labels and Tags”
- “Configuring Security”
- “Starting Up Back Office”
- “Managing Parameters”
- “Scheduling Post-processors”

Loading Labels and Tags

For loading Labels and Tags, 360Commerce provides an SQL script, `ant init_labels`. This script is located in the `backofficeDBInstall.jar` file.

Configuring Security

You can use your own security system, for example, an LDAP server, or you can use the security provided with the Back Office database. The Back Office database requires certain security information, which can either come from another database or be set manually.

Back Office offers more than 300 security roles that control access to different functions. Work with 360Commerce to map your business security roles to the roles Back Office provides.

Back Office also enables you to control workflow through approval permissions, which require that certain tasks scheduled by employees must be approved by other employees before they can take place.

Security roles are listed in the `\applications\backoffice\deploy\backoffice.ear\application.xml` file in the source code control system.

An additional consideration is browser security. When multiple users use the same systems, savvy end-users can use the browser history trail to access data not appropriate for their access levels. Configure the browsers used to access the application to remove access to the history function and the address bar to prevent this.

Starting Up Back Office

Performing some final configuration tasks, such as importing parameters, requires running the application.

To run Back Office:

1. Verify that the application is available in the Application Server environment.
2. Access the application from a browser, using the following URL format:

```
http://<appserver-hostname>:<application port>/backoffice
```

Managing Parameters

Parameters are properties whose values control the behavior of 360Commerce applications. They are passed among the applications as XML files using JMS communication. Both Back Office and Central Office store parameters in their databases (the 360Store database for Back Office, and the 360Enterprise database for Central Office). Back Office can be used to update the parameters of specific registers in the Point-of-Sale system; this includes parameters that are hidden from the Point-of-Sale software. Central Office can be used to update parameters for both Back Office and Point-of-Sale registers. Neither Back Office nor Central Office can determine what parameters are currently in use at a register, but they can distribute new parameters to the registers, overriding older parameters.

There are two types of parameters in 360Commerce applications:

- *Ordinary parameters* (referred to simply as “parameters”) are intended for 360Store applications. Each application in the system treats these parameters differently: Point-of-Sale consumes the parameters, using them to direct its operation, while Back Office both consumes them and distributes them to registers, and Central Office only distributes them. Some parameters change the operation of Back Office.

For more information on these parameters, see the *Parameter Names and Values Addendum* document in the `_resources` directory provided with your application documentation. The initial values for these parameters can come from your deployment of 360Store applications or from your development team. You can create parameter lists for distribution, and manage the values of the parameters, using the Central Office Data Management options or the Back Office parameter maintenance options.

- *Application parameters* (referred to as Back Office parameters or Central Office parameters) modify the operation of the relevant application and are not distributed.

You can use parameters to configure the 360Commerce applications in the following ways:

- Import a new or updated master set of parameters as an XML file
- Create a set of parameters for distribution using the Back Office user interface
- Update Back Office parameters to change the operation of Back Office

You must import an initial set of parameters before you can use the application. In Back Office, distribute parameters from Central Office if it is available or write custom SQL commands to load your initial parameters. 360Commerce provides the `ant load_parameters` SQL script, in the `backofficeDBInstall.jar`, to load parameters.

If you need to change parameter values, use the following procedure.

To change a parameter value:

1. Open the `backoffice.xml` file in a text editor or XML editor. The file contains parameter definitions similar to those in the following sample:

```
<PARAMETER name="MaximumSearchResults" type="INTEGER" final="Y" hidden="Y">
  <VALIDATOR class="IntegerRangeValidator"
    package="com.extendyourstore.foundation.manager.parameter">
    <PROPERTY propname="minimum" propvalue="0"/>
    <PROPERTY propname="maximum" propvalue="9999"/>
  </VALIDATOR>
  <VALUE value="9999"/>
</PARAMETER>
```

2. Change the values in the file. Refer to *Parameter Names and Values Addendum* document in the `_resources` directory provided with your application documentation for a description of the parameters to assist you in determining appropriate values.

Note: You can change the value of a parameter from one valid value to another; however, you cannot change the `<VALIDATOR>` options, such as minimum or maximum values, without corresponding changes in the code.

3. Import the file.

Scheduling Post-processors

Schedule post-processor jobs after installing Back Office.

To schedule regular post-processor jobs within Back Office:

1. Select **Admin > Job Manager**.
2. From the list of import options, select **Available Imports**.
3. From the available imports, click **Schedule** adjacent to the Transaction Post Processor.
Back Office displays the Job Schedule page.
4. Choose **Scheduled**. Back Office displays additional scheduling options.
5. Enter the current date in the Begin Date field.

-
6. Check the Repeating check box.
 7. Leave the No End radio button selected.
 8. Set the Repeating options Daily and Interval.
 9. Enter a run time in the appropriate box.
 10. Click **Add**. The time you entered in step 9 is displayed in the Scheduled Times section at the bottom of the screen.
 11. Click **Next**. The system displays the Notification screen.
 12. Add the e-mail addresses of anyone you want to be notified about the post-processor job.
 13. Click **Next**. The Distribution Summary screen opens, with a summary of the post-processor job displayed in the Task Information box.
 14. Click **Submit Job**. The Distribution Confirmation screen opens.
 15. Click **Done**.